CLAIMS:

1. A method comprising:

verifying security of a boot code associated with a peripheral device by performing a security check on the boot code in accordance with a certificate that describes operation of the boot code; and

executing the boot code based on a result of the security check.

2. The method of claim 1, wherein verifying the security of the boot code includes verifying the boot code via Efficient Code Certification that specifies a process for performing the security check on the boot code as indicated by the certificate.

3. The method of claim 1, wherein the certificate further indicates a type of security check to perform.

4. The method of claim 3, wherein the type of security check comprises one of a security check to enforce type safety, a security check to enforce control flow safety, a security check to enforce memory safety, a security check to enforce stack safety, a security check to enforce device encapsulation and a security check to enforce prevention of specific forms of harm.

5. The method of claim 1, wherein the boot code includes boot firmware.

6. The method of claim 5, wherein the boot firmware conforms to Open Firmware standard IEEE-1275.

7. The method of claim 1, wherein verifying the safety of the boot code occurs inline such that verifying the safety of the boot code occurs in real time prior to executing the boot code.

8. The method of claim 1, wherein the boot code includes boot code defining a device driver to initialize the peripheral device and define an application program interface for accessing and controlling the peripheral device.

9. A method comprising:

generating a boot code for a peripheral device from a program written in a high-level programming language;

gathering information while generating the boot code; and

generating a certificate from information gathered while generating the boot code, wherein the certificate describes operation of the boot code.

10. The method of claim 9, wherein generating the boot code comprises:

compiling the program written in the high-level programming language into a bytecode;

translating the bytecode into a program written in a low-level programming language; and

tokenizing the program written in the low-level language into the boot code.

11. The method of claim 10, wherein gathering information while generating the boot code comprises gathering compilation information while compiling the program written in the high-level language into the bytecode.

12. The method of claim 11, wherein the program written in the high-level language includes a call to a verification application program interface, which provides secure access to the peripheral device.

13. The method of claim 10, wherein the low-level programming language includes Forth.

14. The method of claim 9, wherein the high-level programming language includes one of Java, C++ and Visual Basic.

15. The method of claim 9, wherein the boot code comprises boot firmware.

16. The method of claim 15, wherein the boot firmware conforms to Open Firmware standard IEEE-1275.

17. The method of claim 9, further comprising verifying security of the program written in the high-level programming language prior to generating the boot code, and wherein generating the boot code includes generating the boot code based on the result of verifying the security of the program written in the high-level programming language.

18. A device comprising:

a control unit to verify security of a boot code associated with a peripheral device by performing a security check on the boot code in accordance with a certificate that describes operation of the boot code; and

a memory module whereby the control unit executes the boot code based on a result of the security check.

19. The device of claim 18, wherein the control unit verifies the boot code using principles of Efficient Code Certification.

20. The device of claim 18, wherein the certificate further indicates a type of security check to perform.

21. The device of claim 20, wherein the type of security check comprise one of a security checks to enforce type safety, a security check to enforce control flow safety, security checks to enforce memory safety, security checks to enforce stack safety, security checks to enforce device encapsulation and security checks to enforce prevention of specific forms of harm.

22. The device of claim 18, wherein the boot code includes boot firmware.

23. The device of claim 22, wherein the boot firmware conforms to Open Firmware standard IEEE-1275.

24. The device of claim 18, wherein the control unit verifies the safety of the boot code in real time prior to executing the boot code.

25. The device of claim 18, wherein the boot code defines a device driver to initialize the peripheral device and define an application program interface for accessing and controlling the peripheral device.

26. A device comprising a control unit to generate a boot code for a peripheral device from a program written in a high-level programming language and generate a certificate from information gathered while generating the boot code, wherein the certificate describes operation of the boot code.

27. The device of claim 26, wherein the control unit compiles the program written in the high-level programming language into a bytecode, translates the bytecode into a program written in a low-level programming language, and tokenizes the program written in a low-level language into the boot code.

28. The device of claim 27, wherein the control unit generates the certificate from compilation information gathered by the control unit while the control unit compiles the program written in the high-level language into the bytecode.

29. The device of claim 27, wherein the low-level programming language includes Forth.

30. The device of claim 26, wherein the high-level programming language includes one of Java, C++ and Visual Basic.

31. The device of claim 26, wherein the boot code comprises boot firmware.

32. The device of claim 31, wherein the boot firmware conforms to Open Firmware standard IEEE-1275.

33. The device of claim 26, wherein the program written in the high-level language includes a call to a verification application program interface, which provides secure access to the peripheral device.

34. The device of claim 26, wherein the control unit verifies security of the program written in the high-level programming language prior to generating the boot code and generates the boot code based on the result of the verification of the security of the program written in the high-level programming language.

35. A system comprising:

a peripheral device having a memory module, wherein the memory module stores a boot code and a certificate; and

a computer having a second memory module and a control unit, wherein the control unit retrieves the boot code and the certificate from the peripheral device and executes a verification module that verifies security of the boot code by performing a security check on the boot code in accordance with a certificate that describes operation of the boot code, and

wherein the control unit further executes the boot code based on a result of the security check.

36. The system of claim 35, wherein the control unit verifies the boot code using principles of Efficient Code Certification.

37. The system of claim 35, wherein the certificate further indicates a type of security check to perform.

38. The system of claim 37, wherein the type of security check comprise one of a security check to enforce type safety, a security check to enforce control flow safety, a security check to enforce memory safety, a security check to enforce stack safety, a security check to enforce device encapsulation and a security check to enforce prevention of specific forms of harm.

39. The system of claim 35, wherein the verification module verifies the safety of the boot code in real time prior to executing the boot code.

40. The system of claim 35, wherein the boot code defines a device driver to initialize the peripheral device and to define an application program interface for accessing and controlling the peripheral device.

41. The system of claim 35, wherein the peripheral device comprises one of a graphic device, network controller and storage controller.

42. A system comprising:

a peripheral device having a memory module; and

a control unit to generate a boot code from a program written in a high-level programming language, generate a certificate from information gathered while generating the boot code, and load the boot code and the certificate into the memory module, wherein the certificate describes operation of the boot code.

43. The system of claim 42, wherein the control unit compiles the program written in the high-level programming language into a bytecode, translates the bytecode into a program written in a low-level programming language, and tokenizes the program written in a low-level language into the boot code.

44. The system of claim 43, wherein the control unit gathers compilation information while the control unit compiles the program written in the high-level language into the bytecode.

45. The system of claim 44, wherein the program written in the high-level language includes a call to a verification application program interface, which provides secure access to the peripheral device.

46. The system of claim 42, wherein the control unit verifies security of the program written in the high-level programming language prior to generating the boot code and generates the boot code based on the result of the verification of the security of the program written in the high-level programming language.

47. A computer-readable medium comprising instructions for causing a programmable processor to:

verify security of a boot code associated with a peripheral device by performing a security check on the boot code in accordance with a certificate that describes operation of the boot code; and

execute the boot code based on a result of the security check.

48. The computer-readable medium of claim 47, wherein the instructions for causing the programmable processor to verify the security of the boot code includes instructions to verify the boot code using principles of Efficient Code Certification.

49. The computer-readable medium of claim 47, wherein the certificate further indicates a type of security check to perform.

50. The computer-readable medium of claim 49, wherein the type of security check comprise one of a security check to enforce one of type safety, a security check to enforce control flow safety, a security check to enforce memory safety, a security check to enforce stack safety, a security check to enforce device encapsulation and a security check to enforce prevention of specific forms of harm.

51. The computer-readable medium of claim 47, wherein the boot code includes boot firmware.

52. The computer-readable medium of claim 51, wherein the boot firmware conforms to Open Firmware standard IEEE-1275.

53. The computer-readable medium of claim 47, wherein instruction causing the programmable processor to verify the safety of the boot code includes instructions causing the programmable processor to verify the safety of the boot code in real time prior to executing the boot code.

54. The computer-readable medium of claim 47, wherein the boot code includes boot code defining a device driver to initialize the peripheral device and to define an application program interface for accessing and controlling the peripheral device.

55. A computer-readable medium comprising instructions for causing a programmable processor to:

generate a boot code for a peripheral device from a program written in a high-level programming language; and

generate a certificate that describes operation of the boot code from information gathered while generating the boot code.

56. The computer-readable medium of claim 55, wherein the instructions to generate the boot code comprises instructions to cause the programmable processor to:

compile the program written in the high-level programming language into a bytecode;

translate the bytecode into a program written in a low-level programming language; and

tokenize the program written in a low-level language into the boot code.

57. The computer-readable medium of claim 56, wherein information gathered while generating the boot code, further includes compilation information gathered while compiling the program written in the high-level language into the bytecode.

58. The computer-readable medium of claim 56, wherein the high-level programming language includes Java, C++ and Visual Basic.

59. The computer-readable medium of claim 56, wherein the low-level programming language includes Forth.

60. The computer-readable medium of claim 55, wherein the boot code comprises boot firmware.

61. The computer-readable medium of claim 60, wherein the boot firmware conforms to Open Firmware standard IEEE-1275.

62. The computer-readable medium of claim 55, wherein the program written in the high-level language includes a call to a verification application program interface, which provides secure access to the peripheral device.

63. The computer-readable medium of claim 55, further comprising instruction to cause the programmable processor to verify security of the program written in the high-level programming language prior to generating the boot code and generating the boot code includes generating the boot code based on the result of verifying the security of the program written in the high-level programming language.

64. A method comprising:

generating a boot code in the fcode programming language for a peripheral device from a program written in the Java programming language; and

generating a certificate from information gathered while generating the boot code, wherein the certificate describes operation of the boot code.